Structured Programming

CPSC 110

Drawn from James Tam's material

J. Michael Moore
Programming Structure

- ___________________
- ___________________
- ___________________
- ___________________
Sequence

• Programs are executed _______________ at a time.
• A compound statement _______________ a sequence of statements as a ____________ statement.
Compound Statement

begin
    statement1;
    statement2;
    .
    .
end;
High Level View Of Decision Making For The Computer

Is income below $10,000?

- True
  - Nominal income deduction
  - Eligible for social assistance

- False
  - Income tax = 20%
  - etc.

Is income between $10K - $20K?

J. Michael Moore
Decision-Making In Pascal

• Decisions are questions with answers that are either _____ or ______ (Boolean) e.g., Is it true that the variable 'x' is positive?

• The program branches one way or another depending upon the answer to the question.
If-Then

Question?

- True: Execute a statement
- False: Remainder of the program
If-Then

- Decision-making: checking if a particular condition is true

**Format:**

if (boolean-expression) then

statement;

additional statements;

- **Example:**

if (age >= 18) then

writeln('You are an adult');

writeln('Tell me more about yourself');
Allowable Operands For Boolean Expressions

Boolean-expression:
operand relational-operator operand
OR
boolean-value

Operands:

• integer
• real
• boolean
• char
• const

Can also stand alone as a Boolean expression (it is a boolean value)
Relational Operators For Boolean Expressions

If \((\text{operand relational-operator operand})\) then

<table>
<thead>
<tr>
<th>Pascal operator</th>
<th>Mathematical equivalent</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;</td>
<td>&lt;</td>
<td>Less than</td>
</tr>
<tr>
<td>&gt;</td>
<td>&gt;</td>
<td>Greater than</td>
</tr>
<tr>
<td>=</td>
<td>=</td>
<td>Equal to</td>
</tr>
<tr>
<td>&lt;=</td>
<td>(\leq)</td>
<td>Less than or equal to</td>
</tr>
<tr>
<td>&gt;=</td>
<td>(\geq)</td>
<td>Greater than or equal to</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>(\neq)</td>
<td>Not equal to</td>
</tr>
</tbody>
</table>

J. Michael Moore
If-Then (Simple Statement)

- **Body of if-then consists of a single statement**

- **Format:**
  
  ```
  if (Boolean expression) then
  s1;
  s2;
  ```

- **Example:**
  
  ```
  if (x = 1) then
  writeln('Body of if');
  writeln ('After body');
  ```

J. Michael Moore
If-Then (Compound Statement)

• Body of if-then consists of multiple statements
• **Format:**

  ```
  if (Boolean expression) then
  begin
    s1;
    s2;
    : 
    sn;
  end;
  sn+1;
  ```

**Indicates** ____________________________________________________________

______________________________________________________________

J. Michael Moore
If-Then (Compound Statement)

- Example:

```pascal
taxRate := 0.2;
  if (income < 10000) then
    begin
      writeln('Eligable for social assistance');
      taxCredit = 100;
    end;
  tax = income * taxRate;
```
If-Then: What Gets Executed

• When true, the if executes ________________ which is a ___________ statement.
• __________ statement: the body follows the 'then' and precedes the first semi-colon.
• ________________ statement: the body is enclosed within the begin-end pair as a compound statement.
If-Then-Else

Question?

True

Execute a statement

False

Execute a statement

Remainder of the program

J. Michael Moore
If-Then-Else

• Decision-making with two conditions (true or false)

• **Format:**

```
if (boolean-expression) then
    body of 'if'
else
    body of 'else';
additional statements;
```

- No semi-colon ___________  
  __________________________
  __________________________  
  !!!

- Semi-colon _______________  
  __________________________
  __________________________  
  .

- Note: ____________________  
  __________________________
  __________________________.

J. Michael Moore
If-Then-Else

• Example:

```pascal
if (age >= 18) then
  writeln('Adult')
else
  writeln('Not an adult');
writeln('Tell me more about yourself');
```
If-Then-Else (Simple Statement)

• Body of if-then-else consists of a ________ statement

• Format:

```plaintext
if (Boolean expression) then
    s1
else
    s2;
    s3;
```

---

J. Michael Moore
If-Then-Else (Simple Statement)

• Example:

```plaintext
if (x = 1) then
    writeln('body of if')
else
    writeln('body of else');
writeln('after if-then-else');
```

J. Michael Moore
If-Then-Else
(Compound Statement)

- if (boolean-expression) then
  begin
    s1;
    :
    sn;
  end
else
  No semi-colon (______________________________!)
  begin
    sn+1;
    :
    sn + m;
  end;
  Semi-colon (______________________________)
  end;
  _________________________________!
  sn + m + l;  ________________________________

J. Michael Moore
If-Then
(Compound Statement)

• Example:

```pascal
if (income < 10000) then
    begin
        writeln('Eligible for social assistance');
        taxRate := 0.1;
    end
else
    begin
        writeln('Not eligible for social assistance');
        taxRate := 0.2;
    end;
tax = income * taxRate;
```

J. Michael Moore
Quick Summary: If Vs. If-Else

• If:
  – Evaluates a _________________ (asks a question)
  – If the expression evaluates to _____ then execute the '______' of the _____.
  – If the expression evaluates to _______ then ____________________________________.
  – Use when your program evaluates a ___________ ___________ and code will be ___________ only when the expression evaluates to ________.
Quick Summary: If Vs. If-Else

- If-else:
  - Evaluates a __________________ (asks a question)
  - If the expression evaluates to _____ then execute the '_______' of the ______.
  - If the expression evaluates to _____ then execute the '_______' of the ______.
  - Use when your program evaluates a ______________ _______ and ______ code will ____________ if the expression evaluates to ______ than if the expression evaluates to ______.
Nested Decision Making

- Decision making is dependent.
- The first decision must evaluate to ________ before the ________________ decisions are even ________________ for evaluation.

Question 1?

True

Statement

False

Question 2?

True

Remainder of the program

False

J. Michael Moore
Nested Decision Making

- One decision is made ________________
- ______ decisions must evaluate to ______ before ______ decisions are even __________ for evaluation.
- Format:

```plaintext
if (Boolean expression) then
  if (Boolean expression) then
    inner body
```

- Example:

```plaintext
if (income < 10000) then
  if (citizen = true) then
    writeln('Eligable for social assistance');
tax := income * TAX_RATE;
```

J. Michael Moore
Nested Decision Making: The Dangling Else

if (x > 0) then
if (y > 0) then
writeln('x and y greater than zero')
else
writeln('x is greater than zero');
The Dangling Else Reformatted

if \(x > 0\) then
  if \(y > 0\) then
    writeln('x and y greater than zero')
  else
    writeln('x greater than zero');
Decision-Making With Multiple Alternatives

• if-then
  – Checks a condition and executes the body if the condition is ______

• if-then-else
  – Checks a condition and executes one body if the condition is ______ and another body if the condition is ______

• Approaches for multiple (two or more) alternatives
  – Multiple if’s
  – Multiple else-if’s

J. Michael Moore
Decision Making With Multiple If's

Question?
- True
  - Statement
  - Question?
    - True
      - Statement
      - Remainder of the program
    - False
      - False

J. Michael Moore
Multiple If's: Non-Exclusive Conditions

• _____, ____ or ______ of the conditions may be true (__________________)

• **Format:**

```plaintext
if (Boolean expression 1) then
  body 1;
if (Boolean expression 2) then
  body 2;
:
statements after the conditions;
```

J. Michael Moore
Multiple If's: Non-Exclusive Conditions (Example)

• Example:

```plaintext
if (x > 0) then 
    writeln('X is positive');
if (y > 0) then 
    writeln('Y is positive');
if (z > 0) then 
    writeln('Z is positive');
```

J. Michael Moore
Multiple If's: Mutually Exclusive Conditions

- At most only one of many conditions can be true
- Can be implemented through multiple if's

```plaintext
if (gpa = 4) then
    letter := 'A';
if (gpa = 3) then
    letter := 'B';
if (gpa = 2) then
    letter := 'C';
if (gpa = 1) then
    letter := 'D';
if (gpa = 0) then
    letter := 'F';
```

J. Michael Moore
Decision Making With If, Else-If

- Question?
  - True: Statement
  - False: Question?
    - True: Statement
    - False: Statement

Remainder of the program
Multiple If, Else-If's: Mutually Exclusive Conditions

• Format:

```plaintext
if (Boolean expression 1) then
  body 1
else if (Boolean expression 2) then
  body 2
  :
else
  body n;
statements after the conditions;
```

J. Michael Moore
Multiple If, Else-If's: Mutually Exclusive Conditions (Example)

```pascal
if (gpa = 4) then
  letter := 'A'
else if (gpa = 3) then
  letter := 'B'
else if (gpa = 2) then
  letter := 'C'
else if (gpa = 1) then
  letter := 'D'
else if (gpa = 0) then
  letter := 'F'
else
  writeln('GPA must be one of 4, 3, 2, 1 or 0');
```

J. Michael Moore
Testing Decision Making Constructs

• Make sure that the ______ of each possible ______ executes when it should.

• Test:
  1) Obviously ______ cases
  2) Obviously ______ cases
  3) __________________________ cases
Testing Decisions: An Example

program testDecisions (input, output);
begin
  var num : integer;
  write('Enter a value for num: '); readln(num);
  if (num >= 0) then writeln('Num is non-negative: ', num)
  else writeln('Num is negative: ', num);
end.

J. Michael Moore
Avoid Using Real Values When An Integer Will Do

program testExample;
begin
  var num :  real;
  num := 1.03 - 0.42;
  if (num = 0.61) then
    writeln('Sixty one cents')
  else
    writeln('Not sixty one cents');
end.

J. Michael Moore
The Need For Repetition (Iteration / Loops)

- Writing out a simple counting program (1 – 3).

```pascal
program counting (output);
begin
    writeln('1');
    writeln('2');
    writeln('3');
end.
```

J. Michael Moore
The Need For Repetition (2)

• Simple program but what if ______________ need to be made?
  – The source code must be ______________ and ______________ each time that a __________ is needed.

• What if you need the program to ________________?
Basic Structure Of Loops

1) ________________ the control
   a)Control – typically a variable that determines whether or not the loop executes or not.
2) _______ the control against a __________
3) __________ the __________ of the loop
4) __________________ of the control
Types Of Loops

• Pre-test loops
  1. ____________ control
  2. ______ if a condition is _____ (using the control in some Boolean expression)
     a) If the condition ____________ then continue on with the loop (go to step 3)
     b) If the condition ____________ then break out of the loop (loop ends)
  3. ______ the __________ of the loop
  4. ______ the __________ of the control
  5. Repeat step 2

J. Michael Moore
Types Of Loops

• Pre-test loops

General characteristics

– The body of the loop executes ______ or more times
– Execute the body ______ if the condition is ______ (________ when it becomes ______)
– Examples: while-do, for

J. Michael Moore
Types Of Loops (2)

- **Post-test loops**

1. __________ control (sometimes this step is unneeded because the control is set in the body, step 3)
2. ________ the _________ of the loop
3. ________ the _________ of the control
4. ________ if a condition is _____ (using the control in some Boolean expression)
   a) If the condition ______________ then break out of loop (loop ends)
   b) If the condition ______________ then continue on with loop (go to step 2)

J. Michael Moore
Types Of Loops (2)

• **Post-test loops**

  General characteristics
  - The body of the loop executes ________ or more times
  - Execute the body ______ if condition is ________
    (__________ when it's ________)
  - Examples: repeat-until

J. Michael Moore
Pre-Test Loop: While-Do

• Generally used when the number of times that the loop executes
  ____________________________.
• Format:

  while (Boolean expression) do
    body

• Example:

  var i : integer;
  i: = 1;
  while (i <= 5) do
    begin
      writeln('i = ', i);
      i := i + 1;
    end; (* while *)

  J. Michael Moore
Pre-Test Loop: While-Do

• Can be used for almost any stopping condition. Loop executes as long as the ______________ ______________ is ______.

• Format:
  while (Boolean expression) do
  body

• Example
  var i : integer;
  i: = 1;
  while (i <= 5) do
  begin
    writeln('i = ', i);
    i := i + 1;
  end; (* while *)

J. Michael Moore
Tracing The While Loop
Infinite Loops

• Infinite loops ____________
  (the __________________ is never met).
• They tend to be caused by ______ errors.

To stop a program with an infinite loop in Unix simultaneously press the <ctrl> and the <c> keys

J. Michael Moore
Infinite Loops

• The loop control is ________________

```pascal
var i : integer;
i := 1;
while (i <=10) do
  writeln('i=', i);
i := i + 1;
```

To stop a program with an infinite loop in Unix simultaneously press the <ctrl> and the <c> keys

J. Michael Moore
Infinite Loops

• The loop control is __________________

```pascal
var i : integer;
i := 1;
while (i <=10) do
  writeln('i=', i);
i := i + 1;
```

Notice how the formatting helps us see the problem.

To stop a program with an infinite loop in Unix simultaneously press the <ctrl> and the <c> keys

J. Michael Moore
Infinite Loops

• The updating of the loop control _______
  
  
  var i : integer;
i := 10;
while (i > 0) do
begin
  writeln('i = ', i);
i := i + 1;
end;

To stop a program with an infinite loop in Unix simultaneously press the <ctrl> and the <c> keys

J. Michael Moore